Amendments to the Claims:

Re-write the claims as set forth below. This listing of claims will replace all prior versions and listings, of claims in the application:

Listing of the Claims:

1. (currently amended) A method for static single assignment form dead code elimination, the method comprising:

examining a first instruction off of a worklist in memory, wherein the first instruction includes a previous link and a write mask;

examining at least one second instruction ~~of the worklist,~~ wherein the at least one second ~~instructions are sources~~instruction is a source of the first instruction and wherein each of the at least one second ~~instructions include~~ instruction includes a previous link and a write mask;

determining if any components within a particular field of the at least one second instruction are required ~~for the at least one second instruction;~~ ~~and~~

~~if~~ when no components of the at least one second instruction are required~~live~~, ~~delete~~ deleting the ~~first~~ at least one second instruction from the machine ~~code.~~code; and

when any component of the at least one second instruction is required, adding the at least one second instruction to the worklist in the memory.

2. (original) The method of claim 1 further comprising:

generating the worklist by:

for each of a plurality of instructions, determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instruction to the worklist.

3. (original) The method of claim 2 further comprising:

setting a live bit for each component of the plurality of instructions.

4. (currently amended) The method of claim 2 wherein ~~the~~ each critical instruction is an instruction that generates an export value.

5. (currently amended) The method of claim 2 further comprising:

prior to generating the worklist:

receiving a plurality of instructions;

adding to each instruction ~~a~~ the previous link; and

adding to each instruction thea write mask.

6. (original) The method of claim 5 wherein the write mask is a multi-bit field representing a number of components in a superword register.

7. (original) The method of claim 6 wherein each of the plurality of instructions may be written to the worklist a predetermined number of times, wherein the predetermined number of times is based on the number of components in the superword register.

8. (currently amended) A method for static single assignment form dead code elimination comprising:

receiving a plurality of instructions;

adding to each instruction a previous ~~bit~~link;

adding to each instruction a write mask; and

generating a worklist in memory by:

for each of the plurality of ~~instructions;~~instructions, determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instructions to the worklist in the memory.

9. (currently amended) The method of claim 8 further comprising:

setting a live bit for each component of the plurality of instructions;

examining a first instruction off of the worklist;

examining at least one second instruction in the machine code, wherein the at least one second ~~instructions are sources~~instruction is a source of the first instruction;

determining if ~~all elements~~any component within a particular field of the at least one second instruction ~~are~~ is live ~~for the at least one second instruction~~; and

~~if~~ when no ~~elements~~ components of the at least one second instruction are live, deleting the ~~first~~ second instruction from the worklist.

10. (cancelled)

11. (currently amended) The method of claim 8 wherein ~~the~~ each critical instruction is an instruction that generates an export value.

12. (original) The method of claim 8 wherein the write mask is a multi-bit field representing a number of components in a superword register.

13. (original) The method of claim 12 wherein each of the plurality of instructions may be written to the worklist a predetermined number of times, wherein the predetermined number of times is based on the number of components in the superword register.

14. (currently amended) An apparatus for static single assignment form dead code ~~eliminations~~ elimination comprising:

at least one memory device storing a plurality of executable instructions; and

at least one processor operably coupled to the at least one memory device, operative to receive the plurality of executable instructions such that the at least one processor, in response to the plurality of executable instructions is further operative to:

~~examines~~ examine a first instruction off of a worklist, wherein the first instruction includes previous ~~bit~~ link and a write mask;

~~examines~~ examine at least one second instruction of the machine code, wherein the at least one second ~~instructions are sources~~ instruction is a source of the first instruction and each of the at least one second ~~instructions include~~instruction includes a previous link and a write mask;

~~determines~~ determine if ~~all~~ any ~~components~~ component within a particular field of the at least one second instruction is ~~are~~ live ~~for the at least one second instruction~~; and

~~if~~ when no ~~elements~~ components are live, ~~deletes~~ delete the ~~first~~ second instruction from the machine code.

15. (currently amended) The apparatus of claim 14 wherein the at least one processor ~~further~~ in response to the plurality of instructions executable instructions is further operative to :

~~generates~~ generate the worklist by:

for each of a plurality of instructions, determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instruction to the worklist.

~~sets a live bit for each component of the plurality of instructions.~~

16. (currently amended) The apparatus of claim 15 wherein ~~the~~ each critical instruction is an instruction that generates an export value.

17. (cancelled)

18. (currently amended) The apparatus of claim ~~17~~ 15 further comprising:

a superword register operably coupled to the at least one processor, wherein the write mask is a multi-bit field representing a number of components in the superword register.

19. (currently amended) An apparatus for static single assignment form dead code eliminations comprising:

at least one memory device storing a plurality of executable instructions; and

at least one processor operably coupled to the at least one memory device, operative to receive the plurality of executable instructions such that the at least one processor, in response to the executable instructions is further operative to:

~~receives~~ receive a plurality of instructions;

~~adds~~ add to each instruction a previous link;

~~adds~~ add to each instruction a write mask; and

~~generates~~ generate a worklist by:

for each of the plurality of instructions; determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instructions to the ~~worklist~~worklist;

~~examines~~ examine a first instruction off of the worklist;

~~examines~~ examine at least one second instruction from the machine code, wherein the at least one second ~~instructions are sources~~instruction is a source of the first instruction;

~~determines~~ determine if any ~~elements~~ component within a particular field of the at least one second instruction is ~~are~~ live ~~for the at least one second instruction~~; and

~~if~~ when no ~~elements~~ component is ~~are~~ live, ~~deletes~~ delete the ~~first~~ second instruction from the machine code.

20. (original) The apparatus of claim 19 further comprising:

a superword register operably coupled to the at least one processor, wherein the write mask is a multi-bit field representing a number of components in a superword register.

21. (new) The apparatus of claim 15, wherein the at least one processor in response to the plurality of executable instructions is further operative to set a live bit for each component of the plurality of instructions.

22. (new) The method of claim 9, further comprising:

when any component of the at least one second instruction is live, adding the at least one second instruction to the worklist.

23. (new) The apparatus of claim 14, wherein the at least one processor in response to the plurality of instructions executable instructions is further operative to :

when any component of the at least one second instruction is live, add the at least one second instruction to the worklist.

24. (new) The apparatus of claim 19, wherein the at least one processor in response to the plurality of instructions executable instructions is further operative to :

when any component of the at least one second instruction is live, add the at least one second instruction to the worklist.